AMENDMENTS to the DRAWINGS

No amendments or changes to the Drawings are proposed.

REMARKS

Note: All paragraph numbers referring to our disclosure correspond to the paragraph numbers as published in the pre-grant publication of our application.

Statement Regarding Related Patents and Patent Applications

The following patents and patent applications are related to this patent application:

Serial Number	Docket Number	Status
11/463,131	AUS920010277US2	pending, under final rejections
09/838,377	AUS920010277US1	issued as 7,120,900
09/838,376	AUS920010278US1	issued as 7,086,004
09/931,302	AUS920010428US1	issued as 6,883,007

Rejections under 35 U.S.C. §103(a)

Regarding the rejections of Claims 1 - 15 under 35 U.S.C. §103(a) over newly-cited Xing in view of previously-cited Abir and further in view of previously-cited Feinberg, Applicant respectfully disagrees with the Examiner's reasons for the rejection and requests reconsideration.

Unidirectional versus Multidirectional Web Addresses. Please note that a distinguishing factor of our claims over the cited references is our method's ability to receive a *uni*directional web address (e.g. one that is read entirely in one direction such as left-to-right or entirely from right-to-left) and to convert it in a piecewise manner (e.g. by labels) to a *multi*directional web address (e.g. one that some portions (labels) are read in a first direction, but where some portions (labels) are read in at least a second direction). This supports a hybrid scheme of web addresses, where, for example, the English portions (http:// , .com , /index , etc.) can be maintained in English in a left-to-right reading order, but where other portions may be translated to another language which is read in a right-to-left order. Please see our ¶¶0067 - 0072 for specific disclosure regarding this distinction in our method, and including analysis of the known methods shortcomings (e.g. the Unicode's standard approach, and Natural Language processing's

assumptions which are not applicable to interpretation of domain names).

Natural Language Processing and Domain Names. Applicant respectfully requests the Examiner to consider that natural language processing rules or syntax are not applicable to interpretation of domain names for many reasons, including that fact that "punctuation marks" and delimiter characters are interpreted differently, and that portions of domain names are used for routing and addressing, which do not correspond syntactically to any part of a sentence structure or grammatical construct. For example, please see our disclosure at these paragraphs:

> [0071] Normally, in natural language text processing, this is not a problem given that the two orderings can be distinguished by their physical justification on the screen, either right or left. This factor, however, is not available to domain name displays. When a domain name appears in printed text, there is no generally accepted way to indicate the overall reading direction.
>
> [0072] Nonetheless, some may argue that if the entire domain name is in Arabic. then the label hierarchv should be reversed. The problem in adopting this strategy occurs when the entire domain name is not from the same script, as is the case in this example. The method of the invention provides a more desirable multilingual output (4) as illustrated in FIG. 3, wherein the "ABC" label is a right-to-left language component of the domain name, and the "ibm" and "com" labels are left-to-right components of the multilingual domain name. This output is consistent with the current structure of domain names. In this case the full stop characters are ignored, and the bidirectional algorithm is applied to each of the individual labels of the domain name.
>
> [0073] One might assume that Unicode's Bidirectional Algorithm may still be appropriate if it is run independently on each of the individual labels. This strategy also presents problems, however. The problem with this approach involves the use of the hyphen-minus character "-", U002D. In the Unicode Bidirectional Algorithm, the hyphen-minus is assigned to the European Terminator character class. Unfortunately, this causes the character to behave as if it were an European numeral when adjacent to European numerals, as specified in rule W5 in Unicode Standard Annex #9.
>
> [0074] This behavior may be acceptable when processing natural language, but is unacceptable when processing multilingual domain names. In multilingual domain names, the predominant usage of the hyphen-minus is as white space, and not as an European terminator, as illustrated in FIG. 4. A single domain name label in logical order (40) is presented,

> with the same label shown in display order (41)
> which is the output of the Unicode Bidirectional Algorithm.
> If the hyphen-minus characters are treated as white space
> characters consistent with their use in domain name, the
> third display order (42) is obtained. Evident from this
> example is the fact that the Unicode Bidi algorithm is
> inappropriate for yet another reason for displaying multilingual
> domain names.

Applicant is hereby amending the claims to specifically clarify and point out the conversion method between a unidirectional web address to a multidirectional web address with respect to the reading order of the entire address and portions within the address. Applicant respectfully submits that these aspects of the claims are not taught or suggested by the cited references.

Teachings of the Xing Reference. With regard to the newly-cited Xing reference, Applicant respectfully points out that the key operational feature of Xing's approach is to use a look-up table (e.g. their "translation database) to substitute a non-English URL with an English URL (e.g. their "entry corresponding to a valid and current domain name") (emphasis added by Applicant):

> ABSTRACT:
> Aim of this invention is to place those billions of people on
> this planet who are not English-speaking but who want to
> have, or access, an appropriate domain name, on an equal
> footing with the English speaking populations. An embodiment
> of the invention proposed herein employs a **translation**
> **database which has an entire set of international domain**
> **names, which can be in any suitable coding format, as an**
> **entry corresponding to a valid and current domain name**
> **definition (English-based domain name) or an IP address.**
> This method fits into the current domain name system
> (DNS), system without generating conflicts, such as producing
> an unwanted duplicate domain name (DN). This method
> also has the flexibility, advantageously, to work with any
> language and character set. The DN/ML-DN to IP resolution
> scheme forms the basis of this invention. The ML-DN client
> does not talk to a normal DNS, except in the special case
> when the ML-DN is actually a DN (normal English one).
> The normal DNS dose not understand ML-DN and cannot
> translate a ML-DN request to a normal DN request. The
> ML-DNS translates the ML-DN request to a normal DN
> request. Then either the ML-DNS server or the ML-DN
> client can send the translated ML-DN, i.e. a normal DN, to
> a normal DNS to get the corresponding IP address. The
> choice of server or client is based on considerations of

> optimization of speed and ease of programming. Both approaches can be used.

For further clarification, Applicant respectfully asks the Examiner to consider Xing's Claim 1:

> Claim 1: . . .
> (c) examining the incoming Domain Name to determine if it is a non-English multi-lingual Domain Name; if so, replacing said non-English Domain Name by a known English Domain Name which corresponds to the multilingual non-English Domain Name; and wherein said English Domain Name is sent out to an Internet Domain Name System for resolution on an internet;

Applicant respectfully submits that one of ordinary skill in the art would recognize that for non-English domain names for which there is an already-known translation, Xing's solution would be suitable and satisfactory, likely more efficient to perform than a calculated translation, and would avoid duplication of use of English domain names, as set forth in Xing's advantages and objectives.

However, the secondary and tertiary references do not apply to such scenarios where translations of non-English domain names are known in advance, and as such, do not lend themselves to creation of a look-up table prior to execution of the method. Combining the methods of the secondary and tertiary references into Xing's method would take away Xing's efficiency of operation (by removing the efficient look-up operations), and would eliminate Xing's advantage of avoiding production of duplicate English domain names.

Therefore, combining Xing with the secondary and tertiary references in the manner suggested in the rationale for the rejections would not have been obvious because it would have rendered Xing's invention unsatisfactory for its intended purpose. There can be no obviousness or motivation in such a combination and modification, of course.

Teachings of the Abir Reference. Regarding the use of Abir to overcome the deficiencies of Xing's disclosure regarding detecting standard parts of a URL, inferencing, and reordering as Applicant has claimed, Applicant respectfully incorporates the previous arguments regarding Abir's teachings which were submitted to the Examiner in response to the first Office Action (reply dated Aug. 25, 2005), and in the first Appeal Brief (dated July 13, 2006).

Applicant's claims recited that a URL or web address is broken into "labels" by parsing it at points where "full stop" characters appear. Full stop characters are characters in natural language that often are used to end a sentence, for example, but in URL's and web addresses, are used a delimiters of sorts. For example, the web address:

http://www.anycompany.com

which has two full stop characters in it, the first following "www" and the other preceding "com".

Applicant respectfully points out that Abir's disclosure fails to describe using the period or "dot" character as a delimiter, not does it mention breaking the entirety of the web address into labels between the full stop characters. In another example, consider the web addresses:

http://www.help.ibm.com (1)

and:

http://help.servers.ibm.com (2)

Applicant's invention, as claimed, would break the first address into four labels according to the full stop character placement: (a) http://www, (b) help, (c) ibm, and (d) com. Likewise, appellant's invention, as claimed, would bread the second address into four labels as such: (a) http://help, (b) serverse, (c) ibm, and (d) com.

Please consider these passages from Abir's disclosure:

1. On the fly strictly URL translation from Latin to native language address system. Surfers that arrive at locations that are not a part of the portal address assignment system (that is, any Web site located on the World Wide Web) will also see in the URL box their own native language rather then Internet protocol. When the surfer, whether through a portal or through the use of the desk top based option, arrives at non-system users Web site, the transformation algorithms will operate. For example, the address in the case of the address http://www.healthinsurance.com, the system will isolate the part of the address that comes after "http://www" and before the ".com" (also after the ".com" for sub sites.) The system will then convert the words into the Web surfer's native language using the system's simple English Hebrew dictionary. In this case health insurance. Then the system may reverse the order of the Hebrew words add the Hebrew variation of the "http://www" and the ".com", and display the address in the surfer's native language.

Abir col. 6 lines 14 - 31

1. Conversion Algorithm. Referring now to FIG. 1, there is illustrated the "conversion" algorithm for transforming a conventional resource identifier into a friendly resource identifier. (For purposes of this disclosure of this preferred embodiment of the transformation algorithm, the set of characters of the non-Latin written language used are the Hebrew characters.) In step **100**, standard parts of conventional resource identifiers such as "http://www" "com" and "htm" are identified. In step **102**, the standard parts are converted to well-known Hebrew equivalents such as [illegible] for "http://www" and [illegible] for "com". In step **104**, the remaining parts of the conventional resource identifier is analyzed for words that have identifiable meanings. For example, the words "health" and "insurance" would be recognized in the word "healthinsurance". In step **106**, the Hebrew word בריאות would be substituted for "health" and the Hebrew word ביטוח would be substituted for insurance. In step **108**, the complete Hebrew resource identifier would be produced.

Abir col. 4 lines 22 - 42

Please notice that there is no mention in either of these passages of using the period character "." or a full stop character for parsing, but instead Abir's system "isolates the part of the address that comes after the 'http://www' and before the '.com' ". So, in example number (1) above, Abir would not create four labels as we have claimed, but would "isolate" the string "help.ibm". In the second example, Abir would not create four labels, either, but instead would fail because no "http://www" appears in the address (e.g. 'www' is omitted).

Applicant respectfully submits that, for reference during consideration of these reply remarks, a domain name or Universal Resource Locator ("URL") is defined by those in the industry has having a protocol identifier (e.g. http or https, etc.), a top-level identifier (e.g. .com, .org, .net, etc.), a registered domain server name or second-level identifier, an optional third-level identifier (e.g. www, www2, etc.), zero or more subdomains, zero or more subdirectories, and zero or more resource names.

For example, in the URL:

http://www.support.ibm.com/index.htm

The protocol identifier is "http://" for hypertext transfer protocol, the top-level identifier is ".com", the registered domain server name with extension is "ibm", the subdomain is "support", the third-level identifier is "www", and the resource is the HTML document named "index.htm". Each of these portions of the domain name is separated by a Latin period "." character, except for the protocol identifier and the optional third-level identifier.

It is possible to have sub-subdomains, such as the following where "linux" is the subdomain of the subdomain of "support":

http://www.linux.support.ibm.com/index.htm

Note that a period "." character is used again to delimit the sub-subdomain from the third-level identifier "www" and the subdomain "support".

Abir, however, teaches a first step or phase of converting an English or Latin-based domain name by identifying pre-determined "standard parts" of a URL, such as the strings

"http://www", ".com", or "HTM" (see figure 1 #100, col. 4 lines 29 - 31). These "standard parts" are exchanged for alternate language (e.g. Hebrew in their example) strings of characters (fig. 1 #102, col. 4 lines 31 - 36), which are not compatible with the Internet Protocol or Domain Name Server protocols.

In other words, Abir first converts any top-level portions, third-level portions, file extensions, and protocol identifiers to pre-determined alternate language characters or strings. Note that no parsing of the URL is specified, but just finding of pre-determined "standard portions" is disclosed. These can be referred to as "standard portions" by Abir because there are a finite set of options for these portions of a URL (e.g. http, ftp, www, .com, .org, .edu., .gov, .co, .htm, .php, .jsp, .html, etc.)

Then, Abir teaches treating the entire set of characters which are not "standard parts" as a string to be converted to the alternate language using word-for-word conversion, and letter-for-letter conversion when words are not recognized. Abir discloses reversing the order of words if the alternate language is a right-to-left interpreted language.

Please note that Abir is silent as to maintaining the original order of the subdomains and domain names, and is silent as to using the "." character as a full stop character while independently reordering the characters within each portion between the full stop characters.

When applying conventional natural language translation techniques, a Latin period "." character is typically interpreted as signaling the end of a sentence construct within a paragraph, unless it is immediately followed by a paragraph termination character, such as a hard line feed ("LF") or carriage return ("CR") character. So, for example, if the *words of* the phrase:

"I own a dog. It is a good dog. <CR>"

were re-ordered for right-to-left languages and interpreted using conventional natural language translation techniques, it would appear in the following order:

"dog good a is It. dog a own I."

Notice that the *sentences* reversed order, as well as the words within the sentences. This is a fundamental problem of the Unicode Bidirectional Algorithm ("BIDI") as applied to domain

names, which arises due to the fact that the algorithm was designed to process natural language text (e.g. sentences and paragraphs), not URLs. This is also a problem unrecognized and unsolved by Abir, as Abir is silent regarding processing of the portions *between full stop delimiters*.

For example, using the Unicode BIDI process or Abir process, the following URL:

http://www.applyforaloan.bigbank.com

would be recognized as two sentences, and would be reordered for right-to-left readers as follows (including character reordering):

<A>knabgib.naolarofylppa<B>

where <A> is Abir's substitution for "http://www", and <B> is Abir's substitution for ".com". Note that the reversal of the order of the "sentences" has now made the domain name incorrectly ordered (e.g. "bigbank" became a subdomain, and "applyforaloan" became a domain name).

Applicant's invention, however, first parses the URL by using a full stop character (e.g. ".") as a delimiter between "labels". So, in the example of:

http://www.applyforaloan.bigbank.com

Applicant's invention would find four "labels" in this example URL:

Label_1 = "http://www"
Label_2 = "applyforaloan"
Label_3 = "bigbank"
Label_4 = "com"

The characters *within each label* are then re-ordered according to right-to-left reading order for recognized words in the target alternate language, independent of the content of the

other labels:

Label_1' = “http://www”
Label_2' = “ naolarofylppa”
Label_3' = “knabgib”
Label_4' = “com”

Because Applicant's invention *preserves the original order* (e.g. original *positions*) of the labels (e.g. doesn’t treat them as sentences within a paragraph as the Unicode process does), the proper relationship of the portions of the URL are preserved while the characters within the portions are reversed for right-to-left reading:

http://www.naolarofylppa.knabgib.com

Abir is silent as to parsing the URL into labels using a full-stop character is a delimiter between labels, reordering of the characters *within each label*, and producing a URL having the labels in the original order of the original URL but with the reordered characters within the labels. Appellant's first amendment clarified and specified this difference.

In summary, Abir teaches handling of URLs by separating out “standard portions” and then translating everything in between the standard portions as if it were natural language text, not dividing the URL into labels according to full stop characters as label delimiters, preserving the original order of the labels, and reordering characters within labels, as we have claimed.

Old and Well-Known Art; Feinberg's Teachings. With respect to the holding that parsing text into sections based on detected delimiters was well known in the art of natural language and text processing in the current Office Action, Applicant respectfully traverses this holding and requests reconsideration of this holding.

Please note that Applicant is not merely claiming parsing text into sections based on detected delimiters in text or natural language. Applicant believes that this is an over simplification of the claim element:

breaking . . . the unidirectional World Wide Web address into a plurality of labels delimited by a pre-determined full stop punctuation mark between the labels, the labels having an original label display order as encountered from left to right, the labels containing a plurality of characters wherein each character has a determinate display order or an indeterminate display order, the full stop punctuation mark excluding a hyphen-minus character;

which is specific to the interpretation of URL and web addresses, and is not simply "text" or "natural language" as the terms are defined in their normal usage. URL's and web addresses are not natural language, of course. Applicant respectfully requests consideration of this claim element in its entirety, and in the context of the entire claim.

Applicant respectfully notes that the Feinberg reference was first cited against Applicant's claims in the second Office Action on the merits, and the Applicant responded to the teachings of this reference in the second Appeal Brief filed on May 29, 2007. Applicant respectfully incorporates those remarks into the current response, and respectfully traverses the old-and-well-known holding for the foregoing reasons, and because Feinberg does not provide evidence of such a holding.

Applicant agreed then, and continues to agree, that Feinberg is certainly addressed to natural language processing, but URL's are not "natural language" in the sense that Feinberg addresses natural language. The period characters, or full stop characters, in a URL do not delimit full sentences of "spoken language" (col. 1, line 18).

And, the period delimiter "." is not the same as a "neutral character" such as a hyphen "-", which is the basis on which Feinberg processes natural language (col. 1, line 55 and 66), nor are periods "." interpreted the same way in a URL as they are in spoken language, such as European Terminators and Separators (col. 2, lines 3 - 6).

So, Feinberg is not directed towards handling URLs and converting them appropriately, as the Applicant has claimed, but instead Feinberg converts natural languages in a word processor document (col. 6, lines 14 - 22) looking for neutral characters, such as a dash "-" (col. 6 line 33), and if one is found, the user is prompted by highlighting the questionable text and prompting for input as to the direction the text should read (e.g. right to left, left to right, etc.). (col. 6, lines 34 - 37, col. 7 lines 3 - 18). As such, Feinberg is not a fully automated bi-directional text processing system (as Appellant's claims are fully automatic), but instead is a tool which requires user interaction to complete its task.

Therefore, Feinberg is silent regarding applying any part or form of their invention to URL's, and silent as to an automated method or machine for properly handling bidirectional URLs.

Traversal of Holdings of Old and Well Known

Applicant respectfully traverses each and all holdings of old or well-known art set forth in the Office Action for the foregoing reasons.

Request for Allowance

Applicant respectfully request allowance of the pending claims for the foregoing reasons, and in particular because the cited references fail to teach or suggest the claim elements of converting a unidirectional web address to a multidirectional web address, and of original label display order being preserved while producing bidirectionality of characters within each label.

Respectfully Submitted,

/ Robert H. Frantz /

Robert H. Frantz
U.S. Patent Agent, Reg. No. 42,553
Tel: (405) 812-5613
Franklin Gray Patents, LLC